(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Offenlegungsschrift**

(10) **DE 102 18 835 A 1**

(51) Int. Cl.[7]:
H 04 L 9/10
G 06 K 19/00

(21) Aktenzeichen: 102 18 835.1
(22) Anmeldetag: 22. 4. 2002
(43) Offenlegungstag: 6. 11. 2003

DE 102 18 835 A 1

(71) Anmelder:

Deutscher Sparkassen Verlag GmbH, 70565 Stuttgart, DE; Deutscher Genossenschafts-Verlag eG, 65189 Wiesbaden, DE; Bank-Verlag GmbH, 50933 Köln, DE; VÖB-ZVD Bank für Zahlungsverkehrsdienstleistungen GmbH, 53175 Bonn, DE

(74) Vertreter:

Patentanwälte Ruff, Wilhelm, Beier, Dauster & Partner, 70174 Stuttgart

(72) Erfinder:

Püttmann, Hermann, 70190 Stuttgart, DE; Tix, Regina, Dr., 70469 Stuttgart, DE; Richter, Hans Georg, 53229 Bonn, DE; Kraus, Hans Peter, 50181 Bedburg, DE; Wolter, Thomas, Dr., 53179 Bonn, DE; Borneis, Guido, 60311 Frankfurt, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Verfahren zum Herstellen eines elektronischen Sicherheitsmoduls

(57) Die Erfindung schlägt vor, bei der Herstellung einer Chipkarte einen öffentlichen Schlüssel des späteren Abnehmers der Karte in dem ROM abspeichern. Bei der Initialisierung kann dann die Authentizität der eingebrachten Daten anhand einer Signatur überprüft werden. Fällt die Überprüfung negativ aus, wird die Initialisierung der Chipkarte abgebrochen.

DE 102 18 835 A 1

Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zum Herstellen elektronischer Sicherheitsmodule und die nach diesem Verfahren hergestellten Sicherheitsmodule. Chipkarten, die beispielsweise als Zahlungsmittel oder als Signaturkarte verwendet werden können, müssen nach bestimmten vorgeschriebenen Verfahren so gestaltet werden, dass ein Missbrauch ausgeschlossen wird.

[0002] An der Fertigung einer solchen Chipkarte sind verschiedene Instanzen beteiligt. Zunächst gibt es den Chiphersteller, der also das Kernstück der Chipkarte produziert. Auf den Chip wird dann eine ROM-Maske aufgebracht, die von einem anderen Hersteller geliefert wird. Die ROM-Maske enthält unter anderem das Betriebssystem, das für den Betrieb der Chipkarte erforderlich ist.

[0003] Beim letzten Vorgang der Herstellung der Chipkarte muss diese zunächst initialisiert und anschließend personalisiert werden. Bei der Initialisierung werden die Voraussetzungen geschaffen, Personalisierungsdaten in den Speicherbereich des Chips zu laden. Dabei werden alle global nötigen Daten übertragen und die nötigen Dateistrukturen angelegt.

[0004] Bei der anschließenden Personalisierung werden die individuellen Daten in die Chipkarte eingebracht. Die Karten werden dann von den Abnehmern, beispielsweise kreditwirtschaftlichen Verlagen, an Banken oder direkt an Endkunden geliefert.

[0005] Es muss bei der Personalisierung sichergestellt werden, dass die hierzu gehörenden Daten nicht abgehört werden können. Daher werden die Initialisierung und Personalisierung als getrennte Prozessschritte behandelt und auch an unterschiedlichen Stellen durchgeführt.

[0006] Der Erfindung liegt die Aufgabe zu Grunde, ein elektronisches Sicherheitsmodul, insbesondere eine Chipkarte in sicherheitstechnischer Hinsicht weiter zu verbessern.

[0007] Zur Lösung dieser Aufgabe schlägt die Erfindung ein Verfahren mit den im Anspruch 1 genannten Merkmalen vor. Die Erfindung schlägt ebenfalls ein nach diesem Verfahren herstellbares Sicherheitsmodul vor. Weiterbildungen der Erfindung sind Gegenstand von Unteransprüchen.

[0008] Anhand der Überprüfung der Signatur über bestimmte Daten mit Hilfe des in dem Chip abgespeicherten öffentlichen Schlüssels des Abnehmers der Chipkarte kann sichergestellt werden, ob die Initialisierungsdaten tatsächlich von der richtigen Stelle stammen.

[0009] Anstelle des öffentlichen Schlüssels selbst kann auch, wie von der Erfindung in Weiterbildung vorgeschlagen wird, ein von dem öffentlichen Schlüssel abgeleiteter Hashwert bei der Chipherstellung eingebracht werden.

[0010] Bei dem Hashwert handelt es sich um einen Prüfwert, der es ermöglicht, Änderungen des öffentlichen Schlüssels zu erkennen. Zwei verschiedene öffentliche Schlüssel haben in der Praxis immer einen verschiedenen Hashwert. Aus dem Hashwert ist es jedoch nicht möglich, auf den Schlüssel zu schließen, von dem der Hashwert abgeleitet wurde. Auf diese Weise wird es möglich, bei der Initialisierung zu überprüfen, ob die Initialisierungsdaten tatsächlich von der richtigen Stelle, das heißt dem richtigen Abnehmer der Chipkarte, stammen. Wenn eine Überprüfung ergibt, dass der Hashwert und der öffentliche Schlüssel nicht zusammen passen, wird die Initialisierung abgebrochen.

[0011] Der von dem öffentlichen Schlüssel abgeleitete Hashwert hat den Vorteil, dass er weniger Platz benötigt als der öffentliche Schlüssel selbst.

[0012] In Weiterbildung der Erfindung kann vorgesehen sein, dass der Hashwert von dem Abnehmer der Chipkarte erzeugt und dem Hersteller des Chips und/oder der ROM Maske mitgeteilt wird.

[0013] Es kann vorgesehen sein, dass der zur Berechnung des Hashwerts benutzte Algorithmus oder Angaben darüber, welcher bekannte Algorithmus benutzt wurde, dem Hersteller des Chips und/oder der ROM Maske mitgeteilt und in dem Speicher des Chips mit abgespeichert wird.

[0014] Es ist ebenfalls möglich und liegt im Rahmen der Erfindung, dass der Hashwert von dem Hersteller des Chips und/oder der ROM Maske erzeugt und zusammen mit dem zu seiner Erzeugung genutzten Algorithmus in dem Speicher des Chips abgespeichert wird.

[0015] Bei der Initialisierung kann vorgesehen sein, dass der öffentliche Schlüssel und sein Hashwert eingegeben werden, so dass die Überprüfung durch Vergleich mit dem abgespeicherten Hashwert und dem bei der Initialisierung neu eingegebenen Hashwert erfolgen kann.

[0016] Es ist aber ebenfalls möglich und wird von der Erfindung vorgeschlagen, dass der Hashwert des eingegebenen öffentlichen Schlüssels an Hand des Algorithmus berechnet und das Ergebnis mit dem abgespeicherten Wert einfach verglichen wird. Auch dies ist eine Möglichkeit zur Überprüfung der Korrektheit des eingegebenen öffentlichen Schlüssels.

[0017] Eine weitere Möglichkeit zur Überprüfung kann darin bestehen, dass bei der Initialisierung der Chipkarte der öffentliche Schlüssel und der zur Erzeugung seines Hashwerts benutzte Algorithmus eingegeben werden.

[0018] Wenn ein Chipkartenhersteller mehrere Abnehmer hat, so kann erfindungsgemäß vorgesehen sein, dass in einer Chipkarte für jeden der möglichen Abnehmer ein öffentlicher Schlüssel beziehungsweise sein Hashwert und gegebenenfalls der zu seiner Berechnung erforderliche Algorithmus gespeichert werden. Bei der Initialisierung erfolgt dann die Identifizierung, um welchen Abnehmer es sich handelt, in sonstiger Weise. Die Überprüfung des Hashwerts wird aber in der gleichen Weise durchgeführt, wie sie hierin beschrieben wird.

[0019] Erfindungsgemäß kann zur weiteren Verbesserung der Sicherheit auch vorgesehen sein, dass für einen Abnehmer mehrere öffentliche Schlüssel beziehungsweise Hashwerte für mehrere Schlüssel abgespeichert werden, um z. B. auf diese Weise Schlüssel unterschiedlicher Länge zu verwenden.

[0020] Weitere Merkmale, Einzelheiten und Vorzüge der Erfindung ergeben sich aus der folgenden Beschreibung einer bevorzugten Ausführungsform der Erfindung sowie anhand der Zeichnung. Hierbei zeigen:

[0021] **Fig.** 1 schematisch den Aufbau des Chips einer Chipkarte;

[0022] **Fig.** 2 schematisch den Aufbau des Chips nach Einbringung des Initialisierungsimages;

[0023] **Fig.** 3 die Einbringung des abnehmerspezifischen geheimen Schlüssels;

[0024] **Fig.** 4 das Einbringen der Prüfdaten beim Chipkartenhersteller;

[0025] **Fig.** 5 den Zustand des Chips nach erfolgter Überprüfung.

[0026] Der Chip enthält eine ROM-Maske **1**, die von dem ROM-Maskenhersteller produziert und von dem Chiphersteller in den Chip eingebracht wird. Die ROM-Maske enthält unter anderem das Betriebssystem, das für die weiteren Herstellungsschritte der Betrieb des Chips erforderlich ist.

[0027] Weiterhin enthält der Chip ein EEPROM **2**, das zur Aufnahme von Daten und Programmcode bestimmt ist. Das EEPROM **2** ist in drei Bereiche aufgeteilt, nämlich einen Startbereich **3**, einen Prüfbereich **4** und einen Bereich **5** für Daten und Programmcode.

[0028] **Fig.** 1 zeigt den Zustand beim Chiphersteller, in dessen sicherer Umgebung **6** sein Schlüssel aus einem Speicherbereich **7** auf gesichertem Weg in einen Speicherbereich **8** des Startbereichs **3** des EEPROMs eingeschrieben wird.

[0029] Im Einzelnen gilt dabei folgendes:

Key-Management des Chipherstellers und der Abnehmer des Chipherstellers, z. B. der Verlage:

Der Chiphersteller bringt bei der Chipproduktion die ROM-Maske in die evaluierte Chip-Hardware ein. Die Produktionsumgebung des Chipherstellers ist nach den Vorgaben des Signaturgesetzes für die Produktion von SigG konformen Chips zu evaluieren. Der Chiphersteller bestätigt dem Verlag und dem Chipkartenhersteller, dass nur Chips mit evaluierter Hardware für die Produktion von Signaturkarten-Chips verwendet werden.

ROM-Maske des Chips

[0030] Der ROM-Maskenhersteller erstellt die Betriebssystem- und Anwendungssoftware für die Chipkarte in Form einer ROM-Maske.

[0031] Die ROM-Maske enthält für jeden Abnehmer, beispielsweise einen Verlag, zwei 20 Byte lange Hash-Werte über einen verlagsspezifischen öffentlichen Schlüssel PK-Verlag-Chip sowie die 2 Byte lange Byte-Längen des Modulus und die 3 Byte lange Schlüsselkennung Info-PK-Verlag zu dem jeweiligen PK-Verlag-Chip.

[0032] Jeder Verlag stellt dazu im Vorfeld dem ROM-Maskenhersteller diese Werte für Hash-Wert, Byte-Länge und Info-PK-Verlag zur Verfügung. Zusätzlich erhält der ROM-Maskenhersteller den Modulus zum Nachrechnen des Hash-Wertes.

[0033] Die **Fig.** 1 zeigt die Anordnung der Hashwerte in den Feldern **9** bis **12** der ROM-Maske des Chips, jeder Hash-Wert wird wie oben beschrieben in der ROM-Maske durch die jeweiligen Zusatzinformationen ergänzt.

[0034] Die ROM-Maske wird anschließend vom Evaluator nach den gemäß Signaturgesetz vorgesehenen Sicherheitsanforderungen für die technische Komponente zur Erzeugung und Speicherung des Signaturschlüssels evaluiert und dem Chiphersteller zur Aufbringung auf dem evaluierten Chip übergeben.

[0035] Im Rahmen der Chipherstellung bringt der Chiphersteller den Triple-DES-Schlüssel K-Chip zusammen mit Nebeninformationen zum Schlüssel, das Chippasswort und weitere Daten gesichert in den Startbereich **3** des EEPROMs (Größe in der Regel 64 Bytes) der Chipkarte ein.

[0036] Das Betriebssystem des Chips muss durch geeignete Maßnahmen dafür Sorge tragen, dass der eingebrachte Schlüssel K-Chip und das Chippasswort nicht aus dem EEPROM auslesbar sind und der Startbereich nicht manipulierbar ist. Des Weiteren darf die Einbringung des Chippassworts und des Schlüssels K-Chip nur in den Startbereich des EEPROMS möglich sein.

[0037] Das Betriebssystem des Chips ist so zu gestalten, dass Unterprogrammaufrufe, die vom ROM-Code initiiert werden, um Code im EEPROM-Bereich ansprechen zu können, z. B. immer auf eine oder mehrere entsprechende Sprungadresse(n) **13** im Startbereich des EEPROMs weisen. Diese Sprungadressen **13** enthalten vom Zeitpunkt der Chip-Produktion beim Chiphersteller bis zu der erfolgreichen Ausführung des VERIFY_EEPROM-Kommandos immer eine "RETURN"-Anweisung, d. h. die übrigen EEPROM-Bereiche werden weder direkt noch indirekt zur Ausführung von Code adressiert.

[0038] Die Herstellung des Chips der Chipkarte beim Chiphersteller ist damit abgeschlossen. Die produzierten Chips werden nach der Modularisierung an den Chipkartenhersteller ausgeliefert. Der Chipkartenhersteller erhält Chips für die Chipkarte, die noch nicht abnehmerspezifisch sind, sondern erst bei der Initialisierung einem der beispielsweise vier Abnehmer zugeordnet werden. Dies vereinfacht ggf. die Disposition der vorhandenen Chipmengen beim Chipkartenhersteller und reduziert durch die anfallenden größeren Einkaufsmengen beim Chiphersteller den Stückpreis des Chips.

Schlüsselaustausch mit den Verlagen für K-Chip

[0039] Jeder Chiphersteller bringt seinen K-Chip in das Sicherheitsmodul (S-Box) des Initialisierungstools ein, das dafür beim jeweiligen Abnehmer/Verlag aufgestellt wird. Diese S-Box besitzt u. a. folgende Funktionalität:

Einbringung des chipherstellerspezifischen Schlüssels K-Chip;

Einbringung des verlagsspezifischen Schlüssels K-Chip-Verlag;

Verschlüsselung des eingebrachten K-Chip-Verlag mit dem Schlüssel K-Chip;

Verschlüsselung der Schlüssel für den Prüfbereich des Chips mit K-Chip-Verlag;

Berechnung der Signatur über die Prüfwerte und

Erstellung der entsprechenden Chiffren für die Testkarten.

[0040] Das Kryptogramm über K-Chip-Verlag wird später in die Initialisierungstabelle des Chips der Chipkarte übergeben. Durch den verlagsspezifischen K-Chip-Verlag wird eine klare Abgrenzung der Sicherheitskonzepte der einzelnen Verlage erreicht.

[0041] Den Verlagen darf es nicht möglich sein, durch etwaige Kenntnis eines mit K-Chip verschlüsselten Schlüssels K-Chip-Verlag eines anderen Verlags den Schlüssel K-Chip-Verlag eines anderen Verlags zu benutzen. Da K-Chip chipherstellerspezifisch ist, muss in der S-Box des Initialisierungstools die Funktion "Export des K-Chip" und "Entschlüsseln mit K-Chip" gesperrt sein. Es darf nur möglich sein, mit dieser S-Box den K-Chip-Verlag zu verschlüsseln. Die Funktionen der S-Box dürfen nur nach vorheriger Authentikation (z. B. PIN-Eingabe) des Benutzers gegen die S-Box ausführbar sein.

[0042] Initialisierung und Aufbau der Initialisierungstabelle:

Die Initialisierungstabelle ist ein wesentliches Produktionsmittel für die Chipkarte. Sie kann bei der Herstellung der Chipkarte für die Einbringung identischer Speicherinhalte in alle Chipkarten einer ROM-Maske verwendet werden.

[0043] Der ROM-Maskenhersteller erstellt nach Fertigstellung der Programmierung von Code und Datenstrukturen ein Abbild eines speziellen Zustands des persistenten Speichers des Chips, das sogenannte Initialisierungsimage. Dieses soll in den persistenten Speicher der Chipkartenchips geladen werden, um diese zur Aufnahme von Personalisierungsdaten vorzubereiten. Zum Zweck einer geeigneten und sicheren Einbringung muss das Image für den Chip durch Hinzufügen von Kontroll-, Protokoll- und Prüfinformationen in das Format einer produktiven Initialisierungstabelle für die Initialisierungsanlage transformiert werden.

[0044] Die Initialisierungstabelle enthält nach dem Tabellenkopf Untertabellen mit Kommandofolgen, die der Initialisierer an die Chipkarte übergeben muss, sowie Kommandos und Informationen zur Kontrolle und Protokollierung des Initialisierungsprozesses.

[0045] Die Teile der Initialisierungstabelle, die die Kommandos und die dazugehörigen Kommandodaten enthalten, werden durch den Initialisierer satzweise in die Chipkarte geladen, indem der entsprechende Datensatz, bestehend aus einem Basis-Kommando und den zugehörigen Daten, an

den Chip geschickt wird. Der Chip antwortet mit einem Returncode, der mit dem entsprechenden Returncode in der Kommandotabelle verglichen werden muss. Stimmen beide nicht miteinander überein, so ist der abweichende Returncode zu protokollieren und das Laden der Initialisierungstabelle abzubrechen.

[0046] Die **Fig.** 2 zeigt schematisch den Aufbau des EEPROMS der Chipkarte nach Einbringung des Initialisierungsimages. Der für den Programmcode und die Daten vorgesehene Bereich **5** des EPROMs enthält jetzt eine Sprungtabelle **15**, von der aus eine Verzweigung auf mehrere Speicherbereiche **16** mit Programmcode erfolgen kann. Weiterhin enthält der Bereich **5** einen Bereich **17** für ein Filesystem mit konstanten Dateninhalten.

[0047] Einbringung des verlagsspezifischen K-Chip-Verlag beim Chipkartenhersteller:

Als erste Aktion des Initialisierungsvorgangs beim Chipkartenhersteller wird der verlagsspezifische Schlüssel K-Chip-Verlag eingebracht. Das Kryptogramm des Schlüssels wurde dem Chipkartenhersteller zuvor vom Verlag als Teil der Initialisierungstabelle sicher zur Verfügung gestellt.

[0048] Die Initialisierungsanlage sendet ein Kommando VERIFY_CHIPPWD mit dem verschlüsselten K-Chip-Verlag an den Chip der Signaturkarte. Nach Empfang der Daten vergleicht der Chip die übergebenen Schlüssel-Informationen (VID, KID und KV) mit den im Startbereich gespeicherten Werten. Erkennt der Chip so, dass ihm ein neuer Schlüssel übergeben wurde, entschlüsselt der Chip mit Hilfe des Schlüssels K-Chip das Kryptogramm von K-Chip-Verlag. Im Startbereich des EEPROMS wird der Schlüssel K-Chip durch K-Chip-Verlag ersetzt. Die erfolgreiche Ausführung des Kommandos wird durch Änderung des Chipzustands protokolliert. In **Fig.** 3 ist die Einbringung des Schlüssels K-Chip-Verlag dargestellt. Ähnlich wie bei der Darstellung der **Fig.** 1 erfolgt ein Überspielen des geheimen Schlüssels des Abnehmers aus einem Speicherplatz **18** des Chipkartenherstellers in den Speicherbereich **8**, in dem bislang der geheime Schlüssel des Chipherstellers untergebracht war.

[0049] Beschreibung des Kommandos "VERIFY_CHIPPWD":

Das Kommando VERIFY_CHIPPWD ermöglicht es, entweder den Schlüssel K_{Chip} durch den verlagsspezifischen Schlüssel K_{Chip_Verlag} zu ersetzen (falls L_c = '1E' ist) oder das in den Kommandodaten übergebene Passwort anhand eines Vergleichs mit dem im persistenten Speicher gespeicherten Chippasswort zu verifizieren (falls L_c = '08' oder L_c = '1E' ist). In jedem Fall autorisiert eine erfolgreiche Kommandoausführung die externe Welt zur Ausführung weiterer Kommandos.

[0050] Der Fehlbedienungszähler (FBZ) für das Chippasswort und der Fehlbedienungszähler für den K_{Chip_Verlag} müssen persistent im Startbereich des EEPROM gespeichert werden, damit sie bei einer Stromunterbrechung nicht gelöscht werden. Hat bei der Verwendung des Chippassworts oder des K_{Chip_Verlag} durch das Kommando der jeweilige FBZ den Wert '00', so bricht das Kommando mit Fehlermeldung ab.

[0051] Beim Aufruf des Kommandos wird zunächst die Integrität des Startbereichs mit einer proprietär zu realisierenden Routine geprüft.

[0052] Für den Modus "Chippasswort vergleichen" (L_c = '08') wird folgendermaßen verfahren:

Ist der Wert des Fehlbedienungszählers für das Chippasswort '00', so wird das Kommando mit dem Returncode '69 83' abgebrochen. Ist dieser FBZ nicht '00', dann verifiziert der Chip das in den Kommandodaten übergebene 8 Byte lange Chippasswort CHIPPWD anhand eines Vergleichs mit dem im Startbereich des EEPROM stehenden Chippasswort. Bei einem falschen Wert des Chippasswortes wird der FBZ des Chippasswortes um eins dekrementiert und das Kommando wird mit dem Returncode '63 Cx' abgebrochen. Dabei gibt 'x' den Wert dieses FBZ und somit die Anzahl der weiteren Versuche an, also 'x' = '2', '1' oder '0'.

[0053] Ist der Vergleich erfolgreich, so wird der Fehlbedienungszähler des Chippasswortes auf den Initialwert '03' zurückgesetzt, im flüchtigen Speicher wird ein Flag gesetzt, das die erfolgreiche Verifikation des Chippasswortes signalisiert und das Kommando wird mit der Ausgabe des Returncodes '90 00' beendet.

[0054] Der Kommandoaufruf im Modus L_c = '08' wird z. B. verwendet als Authentikation beim Einbringen der Protokolldaten der Modulherstellung, bei der Personalisierung und – im Fall einer zweigeteilten Initialisierungstabelle – zu Beginn des zweiten Teils der Initialisierungstabelle. Am Anfang der Initialisierungstabelle wird es im Modus L_c = '1E' abgesetzt, da dort ein Schlüsselwechsel vorgesehen ist.

[0055] Im Folgenden werden diese Bezeichnungen verwendet:

VDaten: VID2 || KID2 || KV2 || K_{Chip_Verlag} || '00 00 00 00 00'

MAC(VDaten): Retail-CFB-MAC über VDaten berechnet mit K_{Chip_Verlag} und ICV = '00 . . . 00'

CHIPPWD: ein vom ROM-Maskenhersteller vergebenes und vom Chiphersteller eingebrachtes 8 Byte langes Passwort oder MAC(VDaten), falls bereits ein Schlüsselwechsel zu K_{Chip_Verlag} stattgefunden hat

[K_{Chip_Verlag}]: K_{Chip_Verlag} mit dem K_{Chip} im CBC-Mode mit ICV = '00 . . . 00' Triple-DES verschlüsselt

VID1: ZKA-Herstellerkennung des Chipherstellers für den im Chip enthaltenen K_{Chip}

VID2: ZKA-Herstellerkennung des Verlags für den in den Chip einzubringenden K_{Chip_Verlag}

KID1, KID2: Schlüsselnummer/-ID des entsprechenden Schlüssels

KV1, KV2: Schlüsselversion des entsprechenden Schlüssels

[0056] Für den Modus "eventuell Schlüssel wechseln" (L_c = '1E') wird folgendermaßen verfahren:

Der Chip prüft, ob das Tripel (VID1, KID1, KV1) verschieden ist vom Tripel (VID2, KID2, KV2) und ob das im Startbereich befindliche und zu K_{Chip} gehörige Tripel (VID, KID, KV) entweder gleich (VID1, KID1, KV1) oder gleich (VID2, KID2, KV2) ist. Ist dies nicht der Fall, wird das Kommando mit dem Returncode '64 00' abgebrochen.

[0057] L_c = '1E' mit Schlüsselwechsel:

Falls das Tripel (VID, KID, KV) identisch ist mit (VID1, KID1, KV1), wird der Wert des Fehlbedienungszählers für den Schlüssel K_{Chip} geprüft. Ist er '00', so wird das Kommando mit dem Returncode '69 83' abgebrochen. Ist dieser FBZ nicht '00', dann prüft der Chip die Kommandodaten. Das Kryptogramm [K_{Chip_Verlag}] wird mit K_{Chip} entschlüsselt. Mit dem so erhaltenen K_{Chip_Verlag} wird anschließend der Wert MAC(VDaten) berechnet und mit dem entsprechenden Wert aus den Kommandodaten verglichen. Stimmen die beiden MAC-Werte nicht überein, wird der FBZ des K_{Chip} um eins dekrementiert und das Kommando mit dem Returncode '63 Cx' abgebrochen. Dabei gibt 'x' den Wert dieses Fehlbedienungszählers und somit die Anzahl der weiteren Versuche an, also 'x' = 'F' . . . '0'.

[0058] Stimmt der MAC aus den Kommandodaten mit dem berechneten MAC(VDaten) überein, ersetzt der Chip im Startbereich VID, KID und KV des Schlüssels K_{Chip} durch VID2, KID2 und KV2 und K_{Chip} durch K_{Chip_Verlag} und setzt den zugehörigen FBZ auf '10'. Anschließend wird das 8 Byte lange Chippasswort im Startbereich durch den Wert MAC(VDaten) (=CHIPPWD) ersetzt und der FBZ des

Chippasswortes auf '03' gesetzt. Der Chipzustand wird auf 2 gesetzt. Dann wird ein Flag im flüchtigen Speicher gesetzt, das die erfolgreiche Verifikation des Chippassworts signalisiert und das Kommando mit Returncode '90 00' beendet.

[0059] L_c = '1E' ohne Schlüsselwechsel:

Falls das Tripel (VID, KID, KV) identisch ist mit (VID2, KID2, KV2), wird CHIPPWD im Startbereich mit dem in den Kommandodaten übergebenen Wert MAC(VDaten) verglichen. Diese Prüfung verläuft mit Ausnahme der Änderung des Chipzustands wie im Modus Lc = '08':

Ist der Wert des Fehlbedienungszählers des Chippasswortes '00', so wird das Kommando mit dem Returncode '69 83' abgebrochen. Stimmen die beiden Werte für CHIPPWD nicht überein, wird der FBZ des Chippasswortes um eins dekrementiert und das Kommando wird mit dem Returncode '63 Cx' abgebrochen. Dabei gibt 'x' den Wert dieses FBZ und somit die Anzahl der weiteren Versuche an, also 'x' = '2', '1' oder '0'. Ist der Vergleich erfolgreich, so wird der FBZ des Chippasswortes auf den Initialwert '03' zurückgesetzt und im flüchtigen Speicher wird ein Flag gesetzt, das die erfolgreiche Verifikation des Chippasswortes signalisiert. Der Chipzustand wird auf 2 gesetzt und das Kommando wird mit der Ausgabe des Returncodes '90 00' beendet.

[0060] Außer dem Wechsel von K_{Chip} zu K_{Chip_Verlag} ist es mit diesem Kommando im Modus L_c = '1E' ebenfalls möglich, von einem Verlagsschlüssel K_{Chip_Verlag} auf einen anderen Verlagsschlüssel K_{Chip_Verlag} zu wechseln. Dabei wird im Chip der alte K_{Chip_Verlag} behandelt wie oben beschrieben der K_{Chip}. Hierzu bedarf es der entsprechenden Kommandodaten mit der Chiffre des neuen unter dem alten K_{Chip_Verlag} und dem neuen Chippasswort.

[0061] Funktion des Kommandos:

Überprüfung und Verarbeitung des Chippasswortes, ggf. Schlüsseltausch Eingabelänge: L_c '1E' oder '08'

Kommandodaten:

falls L_c = '1E': VID1 || KID1 || KV1 || VID2 || KID2 || KV2 || [K_{Chip_Verlag}] || MAC(VDaten)

falls L_c = '08': 8 Byte langes Chippasswort CHIPPWD

Returncodes:

'90 00': erfolgreich ausgeführt

'6E 00': ungültiger CLA-Wert

'6D 00': ungültiger INS-Wert

'6A 86' ungültiger Wert in P1 oder P2

'67 00': falsche Länge

'6F 00': allgemeiner Fehler – technisches Problem

'63 Cx': Authentikation gescheitert, 'x' weitere Versuche möglich

'69 83': Authentikation blockiert (Fehlbedienungszähler = '00')

'64 00': Execution Error, State of non-volatile memory unchanged (wird ausgegeben, wenn beim Lesen von Daten inhaltliche Fehler oder Inkonsistenzen festgestellt werden)

[0062] Der weitere Initialisierungsvorgang für den Chip der Signaturkarte erfolgt durch Laden der Initialisierungstabelle in den Chip. Um den höheren Sicherheitsanforderungen einer Signaturkarte zu genügen, ist das Initialisierungsimage gegen Manipulation geschützt.

[0063] Laden des Initialisierungsimages in den Chip:

In der Initialisierungsumgebung des Chipkartenherstellers werden nach Einbringung des verlagsspezifischen Schlüssels K-Chip-Verlag in den Speicherplatz **21** zunächst die Bereichsgrenzen BTAB für das Initialisierungsimage übergeben. Dazu enthält der Teil CTRL_TAB der Initialisierungstabelle das INITIALIZE-Kommando mit dem Parameter BTAB. Der Chip akzeptiert die in BTAB vorgegebenen Werte für die Bereichsgrenzen nur, wenn so der Bereich gegenüber den Vorgaben von BChip im Startbereich des Chips eingeschränkt wird oder gleich bleibt. Der Chipzustand ändert sich anschließend auf "Image laden". Ein späteres Überschreiben von BTAB wird durch den Chip bis zum erfolgreichen Abschluss des VERIFY_EEPROM-Kommandos oder bis zu einer Reinitialisierung ausgeschlossen. Keinesfalls darf es möglich sein, als zu initialisierenden Adressbereich Teile des Start-Bereichs des EEPROMs oder andere Speicherbereiche außerhalb des Code-Daten-Bereichs anzugeben und zu manipulieren.

[0064] Im nächsten Schritt wird der Code-/Daten-Bereich des EEPROMs von der Initialisierungsanlage des Chipkartenherstellers anhand der Initialisierungstabelle wie in **Fig.** 2 dargestellt initialisiert, wobei hier die zu initialisierenden Adressbereiche des EEPROMs vom Betriebssystem auf ihre Lage innerhalb der durch BTAB vorgegebenen Bereichsgrenzen überprüft werden:

Anschließend werden die Daten in den Prüfbereich eingebracht, siehe **Fig.** 4. Neben dem erwähnten Speicherplatz **21** für den verlagsspezifischen Schlüssels K-Chip-Verlag gibt es weitere Speicherplätze **20** und **22** bis **24**, die bei der Initialisierung mit Prüfdaten besetzt werden.

[0065] Die Prüfdaten enthalten (auf der Schnittstelle zwischen Initialisierungsmaschine und Chipkarte):

den mit K-Chip-Verlag verschlüsselten Triple-DES-Schlüssel KINITAB_MAC,

den mit K-Chip-Verlag verschlüsselten Triple-DES-Personalisierungs-Schlüssel KPers,

den mit K-Chip-Verlag verschlüsselten Triple-DES-Personalisierungs-Schlüssel KTransfer,

die Zuordnung Z mit der Halbleiter/ROM-Masken-Kombination, die bei VERIFY_EEPROM gegen eine vorhandene Eintragung im Startbereich des Chip-EEPROMs geprüft wird,

den MAC über den Code/Daten-Bereich,

die Zusatzinformation Info-PK-Verlag zum PK-Verlag-Chip,

den öffentlichen Schlüssel PK-Verlag-Chip,

die verlagsspezifische Kennzeichnung der Initialisierung und

die Signatur über den Hash-Wert von KINITAB_MAC || KPers || KTransfer || Z || BTAB || MAC (Image) || Info_PK_Verlag verlagsspezifische Kennzeichnung der Initialisierung.

[0066] Danach erfolgt eine Überprüfung, ob die Inhalte des Code/Datenbereichs **5** des EEPROMS **2** der Chipkarte authentisch sind. Dazu wird ein entsprechendes Kommando an den Chip geschickt. Dies führt zu folgenden Vorgängen:

Das Betriebssystem des Chips bildet zunächst den Hash-Wert über den im Prüf-Bereich des Speichers gespeicherten öffentlichen Schlüssels PK-Verlag-Chip und vergleicht diesen mit dem über Info-PK-Verlag referenzierten, in der ROM-Maske hinterlegten Hash-Wert. Stimmt der berechnete Hash-Wert mit dem zugehörigen, in der ROM-Maske vorhandenen Hash-Wert überein, ist der im Prüf-Bereich gespeicherte PK-Verlag-Chip authentisch.

[0067] Dann prüft der Chip mit dem Schlüssel PK-Verlag-Chip die Signatur über die Prüfdaten P = (KINITAB_MAC || KPers || KTransfer || Z || BTAB || MAC über das Initialisierungsimage || Info_PK_Verlag || verlagsspezifische Kennzeichnung der Initialisierung). Dazu wird zunächst der Hash-Wert (SHA-1) über P gebildet und anschließend mit dem Hash-Wert, der sich durch RSA-Public-Key-Verschlüsselung der Signatur über P mit Hilfe des PK-Verlag-Chip ergibt, verglichen. Stimmen die beiden Hash-Werte überein, sind insbesondere der eingebrachte KINITAB_MAC und der MAC über den EEPROM-Inhalt des Code/Daten-Bereichs authentisch. Die Prüfdaten werden nur akzeptiert, wenn Z mit dem entsprechenden Kennzeichen der Chipherstellerdaten im Startbereich übereinstimmt.

[0068] Anschließend berechnet der Chip mit dem Schlüssel KINITAB_MAC unter Verwendung der Bereichsgrenzen BTAB den MAC über den Code-/Daten-Bereich des EEPROMs (inkl. der Protokolldaten für die Chipherstellung (Byte **1–3**, **8–9** und **14**), für die Initialisierung (ohne die ersten 16 Byte) und für die Personalisierung) und vergleicht diesen mit dem im Prüfbereich gespeicherten MAC. Stimmen beide MACs überein, ist nachgewiesen, dass das EEPROM korrekt initialisiert wurde und das eingebrachte Initialisierungsimage authentisch ist.

[0069] Nach erfolgreicher Überprüfung ändert der Chip seinen Zustand in OK. Nun kann von dem Betriebssystem die in der Sprungadresse **13** gespeicherte RETURN – Anweisung durch die Adresse der Sprungtabelle **15** im Code/Datenbereich **5** des EEPROMS **2** ersetzt werden. Damit werden die Unterprogrammaufrufe des ROM Codes nicht mehr gesperrt, sondern über die Sprungtabelle oder einen anderen Mechanismus an den entsprechenden Programmcode im entsprechenden Bereich **5** des EEPROMs **2** adressiert. Der Programmcode in diesem Bereich ist damit für das Betriebssystem verfügbar und ausführbar. Dies ist schematisch in **Fig.** 5 dargestellt.

[0070] Anschließend kann dann die Personalisierung durchgeführt werden. Die bisherige strikte organisatorische Trennung von Initialisierungs- und Personalisierungsumgebung muss für die Produktion der Chipkarte nicht beibehalten werden, da die Schlüssel verschlüsselt in die Chipkarte eingebracht werden.

Patentansprüche

1. Verfahren zum Produzieren von elektronischen Sicherheitselementen, insbesondere Chipkarten, mit folgenden Verfahrensschritten:
   - 1.1 mindestens ein Prüfwert, insbesondere ein öffentlicher Schlüssel des Abnehmers der Chipkarte, wird bei der Chipherstellung in einem Speicherbereich des Chips gespeichert,
   - 1.2 bei der Initialisierung des elektronischen Sicherheitselements wird ein gegebenenfalls addressierbarer Prüfwert verwendet,
   - 1.3 mit Hilfe des in der Chipkarte gespeicherten Prüfwerts wird die Authentizität von bei der Initialisierung eingebrachten Daten geprüft,
   - 1.4 bei einem negativen Ausgang der Überprüfung wird die Initialisierung abgebrochen.

2. Verfahren nach Anspruch 1, bei dem anstelle des öffentlichen Schlüssels des Abnehmers der Chipkarte ein von diesem abgeleiteter Hashwert in den Speicherbereich des Chips eingebracht wird.

3. Verfahren nach einem der vorhergehenden Ansprüche, bei dem der öffentliche Schlüssel und/oder der Hashwert von dem Abnehmer der Chipkarte erzeugt und dem Hersteller des Chips und/oder der ROM Maske mitgeteilt wird.

4. Verfahren nach Anspruch 3, bei dem der zur Berechnung des Hashwerts benutzte Algorithmus dem Hersteller des Chips und/oder der ROM Maske mitgeteilt und in dem Speicher des Chips mit abgespeichert wird.

5. Verfahren nach Anspruch 2 oder 3, bei dem der Hashwert von dem Hersteller des Chips und/oder der ROM Maske erzeugt und zusammen mit dem zu seiner Erzeugung benutzten Algorithmus in dem Speicher des Chips abgespeichert wird.

6. Verfahren nach einem der Ansprüche 1 bis 5, bei dem der Hashwert des eingegebenen öffentlichen Schlüssels an Hand des Algorithmus neu berechnet und das Ergebnis mit dem abgespeicherten Hashwert verglichen wird.

7. Verfahren nach einem der vorhergehenden Ansprüche, bei dem bei der Initialisierung der öffentliche Schlüssel beziehungsweise Hashwert und der zur Erzeugung seines Hashwerts benutzte Algorithmus angegeben werden.

8. Verfahren nach einem der vorhergehenden Ansprüche, bei dem bei mehreren möglichen Abnehmern der Chipkarte für jeden Abnehmer ein öffentlicher Schlüssel bzw. ein Hashwert und/oder der Algorithmus zu seiner Erzeugung gespeichert wird.

9. Verfahren nach einem der vorhergehenden Ansprüche, bei dem für einen Abnehmer der Karte mehrere öffentliche Schlüssel beziehungsweise Hashwerte für mehrere öffentliche und/oder geheime Schlüssel abgespeichert werden.

10. Sicherheitsmodul, enthaltend einen Chip mit einer ROM Maske und einem EEPROM, wobei in dem ROM ein Hashwert des öffentlichen Schlüssels des Abnehmers der Chipkarte oder der öffentliche Schlüssel selbst abgespeichert und das Betriebssystem derart ausgelegt ist, dass nur bei erfolgreicher Signaturprüfung unter Verwendung des öffentlichen Schlüssels des Abnehmers der Chipkarte die Initialisierung möglich ist.

11. Sicherheitsmodul nach Anspruch 10, bei dem in dem ROM auch Angaben über den zur Berechnung des Hashwerts verwendeten Algorithmus abgespeichert sind.

12. Sicherheitsmodul nach Anspruch 10 oder 11, bei dem bei mehreren möglichen Abnehmern der Chipkarte für jeden Abnehmer ein Hashwert und/oder ein Algorithmus zu seiner Erzeugung abgespeichert ist.

13. Sicherheitsmodul nach einem der Ansprüche 10 bis 12, bei dem für einen Abnehmer der Karte mehrere öffentliche Schlüssel beziehungsweise Hash-Werte für mehrere öffentliche Schlüssel abgespeichert sind.

Hierzu 5 Seite(n) Zeichnungen

- Leerseite -

Nummer: **DE 102 18 835 A1**
Int. Cl.[7]: **H 04 L 9/10**
Offenlegungstag: 6. November 2003

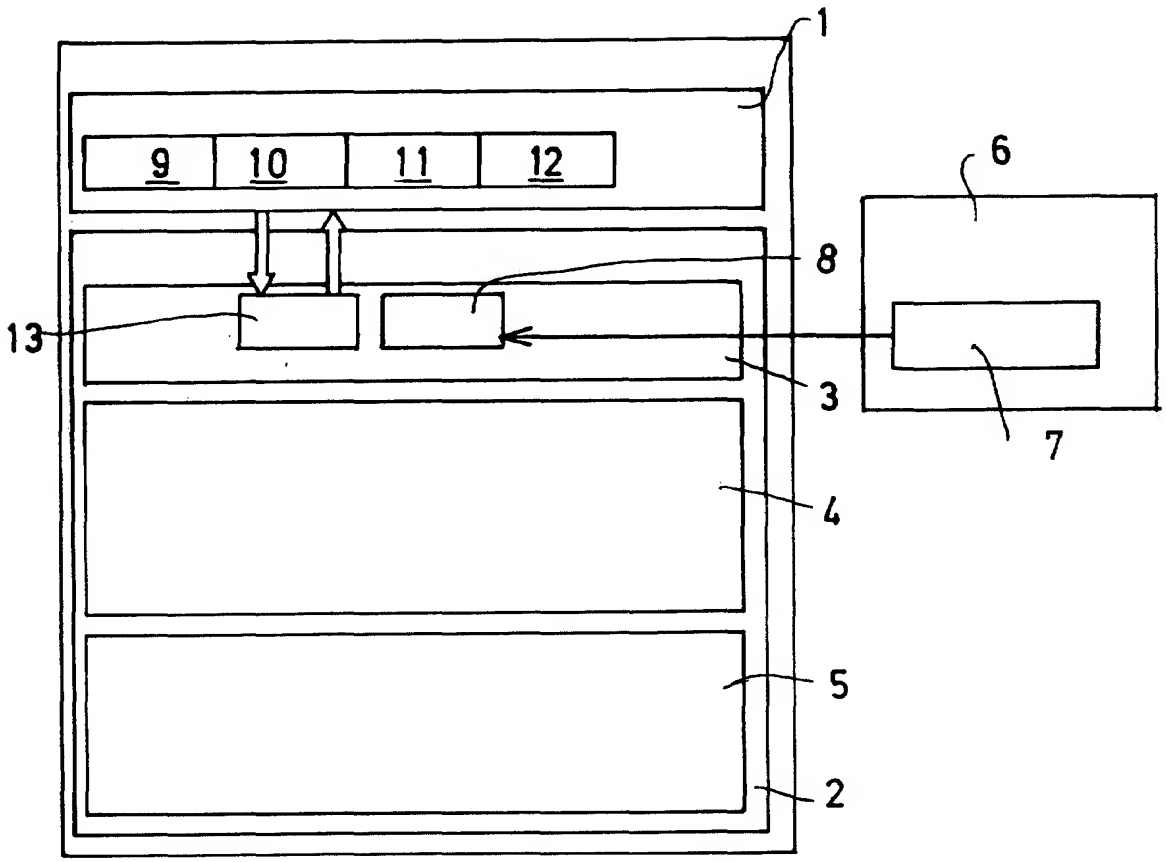

FIG. 1

Nummer: **DE 102 18 835 A1**
Int. Cl.[7]: **H 04 L 9/10**
Offenlegungstag: 6. November 2003

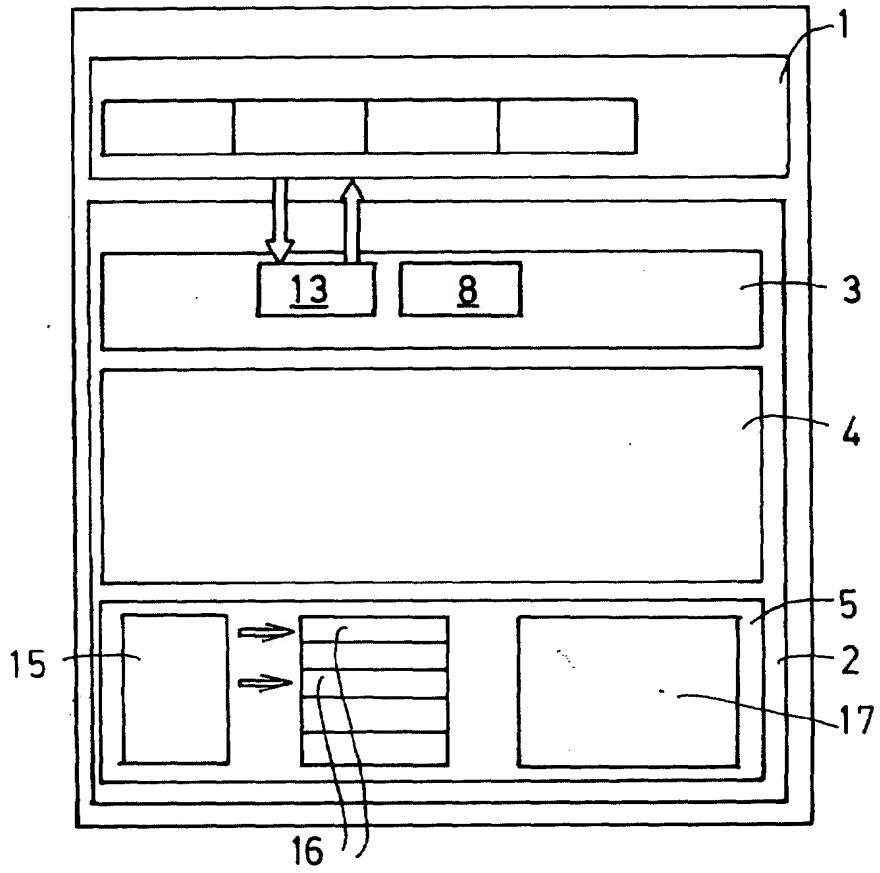

FIG. 2

Nummer: **DE 102 18 835 A1**
Int. Cl.[7]: **H 04 L 9/10**
Offenlegungstag: 6. November 2003

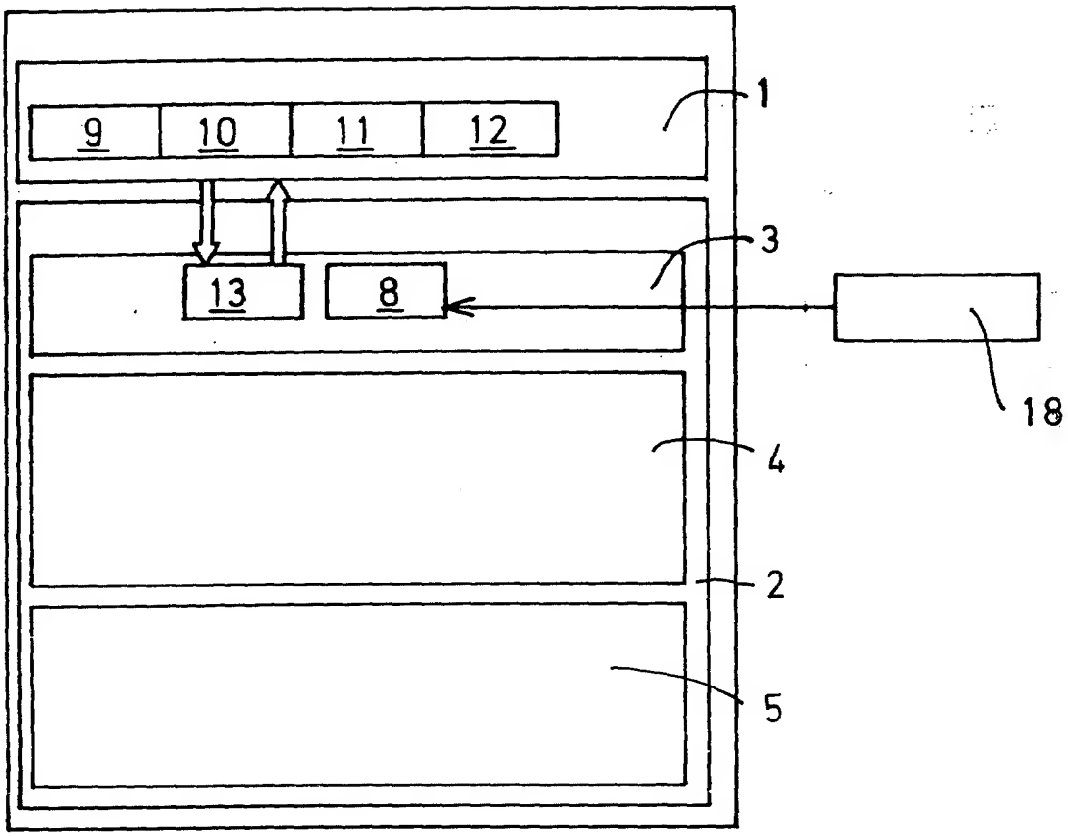

FIG 3

Nummer: **DE 102 18 835 A1**
Int. Cl.[7]: **H 04 L 9/10**
Offenlegungstag: 6. November 2003

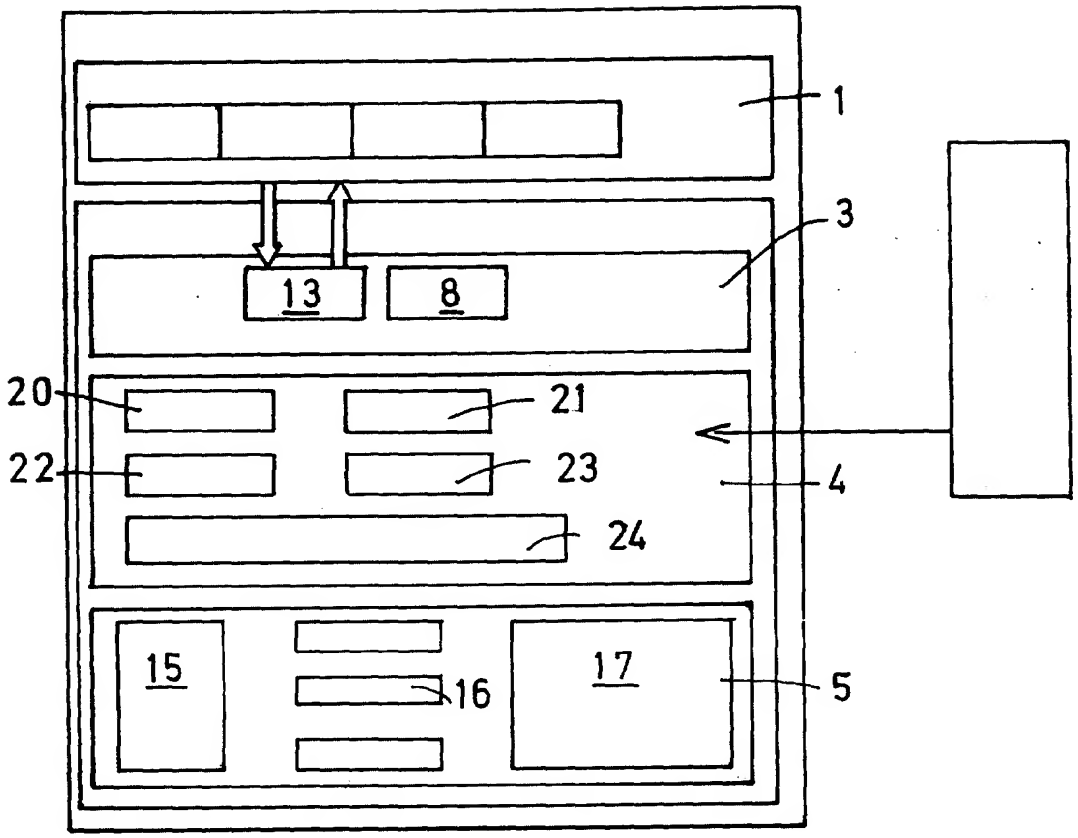

FIG. 4

Nummer: **DE 102 18 835 A1**
Int. Cl.[7]: **H 04 L 9/10**
Offenlegungstag: 6. November 2003

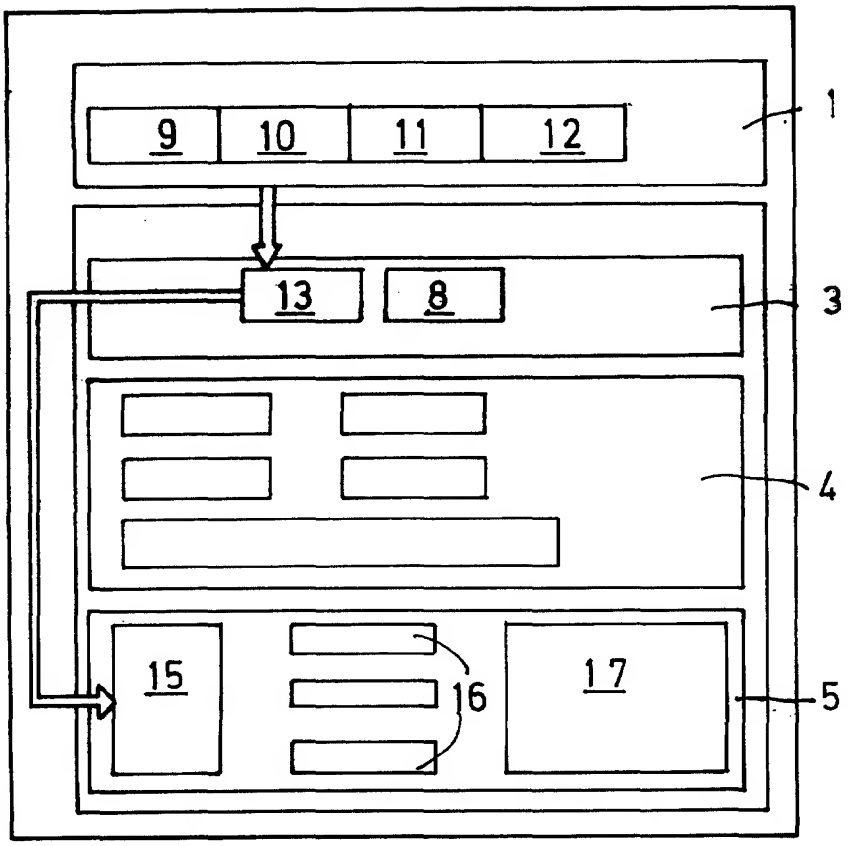

FIG. 5